

### **REMARKS**

Reconsideration and allowance in view of the foregoing amendment and the following remarks are respectfully requested. Claims 1, 36, 38, 44, 46 and 52 are amended. No new matter is added.

#### **Rejection of Claims 29-35, 37-43, 45-51 and 53 Under 35 U.S.C. §103(a)**

The Office Action rejects claims 29-35, 37-43, 45-51 and 53 under 35 U.S.C. §103(a) as being unpatentable over Kusano (U.S. Patent No. 5,796,397) ("Kusano") in view of Song et al. (U.S. Patent No. 6,061,711) ("Song et al."). Applicants respectfully traverse this rejection and submit that the clarifying amendments above as well as additional arguments below shall make it clear that these claims are patentable over the cited art.

First, Applicant respectfully submits that one of skill in the art would not have sufficient motivation to combine Kusano with Song et al. In the Final Office Action regarding the rejection of claim 29, in paragraph 7, the Office Action concedes that Kusano does not specifically teach context switching in claim 29. However, the Office Action asserts that Song et al. teach storing a program state suspending execution of a first program and executing the selected one or plurality of programs citing portions of the Song et al. reference. The Office Action asserts that the reason for the obviousness of the combination is that the teachings of Song et al. would enhance the task switching capability of Kusano by providing the ability to switch between different programs and to return to the point where the user left off in the currently activating program citing the abstract of Song et al. Applicant respectfully submits that when the teachings of the prior art are adequately analyzed for their suggestive power relative to being combined with one another that one of skill in the art would not have sufficient motivation to combine these references.

To establish a *prima facie* case of obviousness, the Examiner must meet three criteria. First, there must be some motivation or suggestion, either in the references themselves, or in

the knowledge generally available to one of ordinary skill in the art, to combine the references. Second, there must be a reasonable expectation of success, and finally, the prior art references must teach or suggest all the claim limitations. The Examiner bears the initial burden of providing some suggestion of the desirability of doing what the inventor has done. "To support the conclusion that the claimed invention is directed to obvious subject matter, either the references must expressly or impliedly suggest the claimed invention or the examiner must present a convincing line of reasoning as to why the artisan would have found the claimed invention to have been obvious in light of the teachings of the references." MPEP 2142.

If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious. *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959). MPEP 2143.01..

Furthermore, if the examiner determines there is factual support for rejecting the claimed invention under 35 U.S.C. 103, the examiner must then consider any evidence supporting the patentability of the claimed invention, such as any evidence in the specification or any other evidence submitted by the applicant. The ultimate determination of patentability is based on the entire record, by a preponderance of evidence, with due consideration to the persuasiveness of any arguments and any secondary evidence. *In re Oetiker*, 977 F.2d 1443, 24 USPQ2d 1443 (Fed. Cir. 1992). The legal standard of "a preponderance of evidence" requires the evidence to be more convincing than the evidence which is offered in opposition to it. With regard to rejections under 35 U.S.C. 103, the examiner must provide evidence which as a whole shows that the legal determination sought to be proved (i.e., the reference teachings establish a *prima facie* case of obviousness) is more probable than not. MPEP 2142.

The test for obviousness is what the combined teachings of the references would have suggested to one of ordinary skill in the art, and all teachings in the prior art must be considered to the extent that they are in analogous arts. Where the teachings of two or more prior art references conflict, the examiner must weigh the power of each reference to suggest solutions to one of ordinary skill in the art, considering the degree to which one reference might accurately discredit another. *In re Young*, 927 F.2d 588, 18 USPQ2d 1089 (Fed. Cir. 1991). MPEP 2143.01.

The mere fact that references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination. *In re Mills*, 916 F.2d 680, 16 USPQ2d 1430 (Fed. Cir. 1990).

Applicant respectfully submits that one of skill in the art would not be motivated to combine these references for several reasons. First, Applicant notes that Kusano teaches specifically the process of changing a current application to a new application during the inputting of data for the application through a composite input screen and the effective use of the input data between the two applications. Also noted in the Abstract is that two kinds of application programs are concurrently stored in the storage means such as a ROM and every application node that has respective input means is stored in an adapted record structure in other storage means such as RAM. Also as noted in column 1, lines 5-14, of Kusano is how their invention relates to an information processing device which is capable of operating with a number of applications by freely changing a current application mode to another mode during the operation of inputting data or correcting the input data in the current mode. This allows for the transferring of the inputted or corrected data into the new mode. Further, in column 1, starting at line 57, Kusano notes that a conventional pocket computer requires the user to previously designate which application is used and then to enter data for the application. To change the application for which the input data shall be used or registered during the input data shall be used or registered during the input operation, the user must

perform a number of steps including storing, temporarily storing a character string into a location such as clipboard interrupting or finishing the data input operation, switching over to another application, bringing into the state for inputting data for the actuated application and pasting the character string or data stored on the clipboard. Therefore, the context of Kusano is highly user interactive in that it specifically is designed to enable multiple applications to be able to receive input data or corrected input data while a user is interacting with one or more of the applications.

In contrast to the context of Kusano, Song et al. teach a method which is applicable in a multi-tasking computing environment in which one program is halted and context switched out so that a processor may context switch in a subsequent program for execution. Song et al. primarily deal with processor state information and how to handle to the storage and retrieval of processor state information as one program is halted and a replacement program is executed. The Abstract in Song et al. discuss how an application programmer chooses points within the programming of an application to locate context appropriate approved locations for context switching will preserve a minimal portion of the processor information that is needed when a context switch is executed. Song et al. introduce their invention, for example, in column 1, starting at line 13 by explaining that multi-tasking generally involves the con-current execution of several programs by a computer by time slicing the program execution. Computer users have the impression that the programs are executed in parallel when in actuality the computer switches between the programs. When program execution is time sliced, a program runs for a period of time before being "context switched out" to run another program. Song et al. further explain that when resuming a context switched out program the program should resume execution at the precise location where execution previously ended. This location is determined by the programmer, not the user. Song et al. further note that the timing of context switching is transparent to the programmer of the context switched out program because from the perspective of that programmer, context

switching occurs at arbitrary times unless an arbitrary location is in the program. In response to the problems identified by Song et al. of context switching in the multi-tasking environment, they introduce their invention to reduce the processor time needed to context switch between programs by identifying appropriate points within the program at which a context switch can occur which can reduce the amount of processor state information needed for successfully resuming the context switched out program. Thus, Song et al. illustrate in FIG. 7 in which various Vector Conditional Context Switch (VCCS) points are located within the flow of the logic of the program which identified the approved places in which a context switch can occur. The Office Action identified column 2, lines 15-20 of Song et al. as identifying information associated with the context switching of programs. What this portion identifies is where in the source code of a program a programmer may insert these VCCS instructions such that the advantageous point of when a context switch would occur is identified.

With this background in mind, one reason that one of skill in the art would not have sufficient motivation or suggestion to combine these references is that Song et al. clearly deal with the multi-tasking computing system environment which is transparent to the user. The environment of the invention of Song et al. relates to how a computer processor processes multiple applications in a time slicing program execution method. Thus, from the stand point of the user, the time slicing process is transparent. The applications run and the time slicing program execution with the invention of Song et al. merely processes without the user's knowledge the context switching that occurs. Applicant notes that Figure 5 of Song et al. shows the multi-media library module, the MSP real time kernel and vector processor labeled 502 which is the module in which the processor in which the invention of Song et al. applies. Notice how this is quite removed from the application in content 506 and the user interface. Kusano's disclosure presents a contrast to the processor based invention of Song et al. Kusano's invention is purely user interface driven as can be seen in Figures 3 and 5-10.

Other examples are shown in Figures 11 through 16B in which the disclosure of Kusano relates directly to a user interaction with a keyboard or with touch sensitive screens and as introduced above Kusano clearly must relate to the method and process by which a user would input data or correct data. Accordingly, Applicant would respectfully submit that the cited prior art references are non-analogous prior art with respect to each other. They are not in the same field and one of skill in the art would not likely reference the teachings of one or the other for further innovation in the context of either of these references. This again is made clear by the fact that Song et al. has absolutely nothing to do with how a user interfaces with any application and Kusano has everything to do with the user interface.

A second reason why Applicant would respectfully submit that the analysis in the Office Action regarding the obviousness to combine these references is inappropriate is that the basis upon the Office Action concludes that it would be obvious to one of skill in the art to combine these references is technically incorrect. Again the Office Action teaches that the Song et al. teachings would enhance the task switching capability of Kusano by providing the ability to switch between different programs and to return to the point where the user left off in the currently activated program. However, the teachings of Song et al. would make absolutely no difference to the task switching capability of Kusano. Because the teachings of Song et al. as discussed above are entirely transparent to the user in terms of the multitasking that occurs in time slicing program execution Applicant submits that the user of a device according to the Kusano reference would see no difference if the teachings of Song et al. were incorporated into Kusano. This means, in other words, Song et al. does not provide "the ability to switch between different programs then to return to the point where the user left off in the currently activating program." As highlighted in Figure 7 of Song et al., the vector conditional context switch (VCCS) taught in Song et al. are inserted at various points by a programmer into the application program. The VCCS switches have nothing to do with a point where a user left off in a currently activating program. In other words, if a user in

Kusano has several applications open and is interacting with several applications the teachings of Song et al. may provide within the programming of those applications VCCS points wherein those programs may operate in a time sliced programming execution manner as it taught in Song et al. but all of that processing will be transparent to the user and does not provide the ability of switching between the different programs and returning to the point where the user left off in the currently activated program.

Thus, Applicant submits that another reason why one of ordinary skill in the art would not be motivated to combine these references is that the manner in which the Office Action characterizes the enhancement achieved by combining these references ignores the lack of user interaction from the Song et al. and therefore the benefit identified is simply not achieved by combining these references without further inventive and non-obvious modifications which would not be suggested to one of skill in the art.

Accordingly, Applicant respectfully submits that it is not obvious to combine these references and accordingly the claims 29-30, 32-33, 35-39, 41-42, 44-47, 49-50 and 52-53 are patentable and in condition for allowance.

Furthermore, Applicant respectfully submits that even if combined the combination of references fails to teach each claim limitation. We turn first to amended claim 29. Claim 29 recites storing a program state associated with a display status when the first user request was received of the first program context packet. As discussed above, Song et al. focus exclusively on context switching at the processor level. The context switching involves saving the state of the program being context switched out prior to the processor loading and running another program. Applicant respectfully submits that because the VCCS switches being placed within the source code of the applications as is taught in Song et al. that this reference fails to teach storing of program status associated with a display means when the first user request was received. This is not taught by Song et al. for several reasons. First, as mentioned above, there is nothing in Song et al. that teaches or suggests that the processor

state information has anything to do with the associated display status. Next, the stored processor information taught in Song et al. is performed based on the location of these VCCS points within the program and not based on when a first user request was received. Finally, the example discussed in Song et al. in column 9, lines 30-65 highlight further that the program state that Song et al. does not teach storing a program state associated with a display status when the first user request was received because as explained by Song et al. the location of the VCCS instruction in the program is meant to reduce the amount of processor state information that is stored. In the example, they discuss how if an add instruction is involved that is supposed to determine the sum of 50 arguments and the application program is context switched out prior to determining of some of the arguments the processor state information necessarily must restore this application program including all 50 arguments. However, the programmer can put the VCCS instruction in following the completion and adding of the sum determined such that only sum must be included as part of the processor stated information. Accordingly, Song et al. actually teach away from the present invention by requiring a delay in the execution of a VCCS instruction based on an effort to reduce the amount of processor state information. Accordingly, a request may come for a context switch but the program will not allow a context switch until the next appropriate VCCS instruction is identified within the flow of the program.

In contrast to this delay, Applicant submits that claim 29 recites storing this program state associated with a display means when the first user request was received and storing that program state in a context packet. Accordingly, for this reason Applicant submits that Kusano and Song et al. fail to teach each limitation of claim 29 and therefore this claim is patentable and in condition for allowance.

Yet another reason exists why amended claim 29 is not taught by the combination of these references. Claim 29 recites upon receiving a second user request, suspending the execution of the selected program and resuming the execution of the first program with the



associated display status based on the context packet. Applicant respectfully submits that clearly Song et al. do not teach such a user interaction requesting context switching because it teaches context switching by a time slicing program execution process. Accordingly, Song et al. does not teach a first user request and furthermore clearly does not teach a second user request the causes the suspension of the execution of the selected program and resuming of the execution of the first program with the associated display status based on the context packet. According, for these additional reasons, Applicant submits that claim 29 is patentable and in condition for allowance. Claims 31-35 depend from claim 29 and recite further limitations therefrom. Claim 36 is restored into its previous state and also allowable.

The limitation previous added to claim 29 from claim 26 is removed from claim 29 and added back into claim 36 and therefore Applicant submits that this claim is patentable and in condition for allowance as well as claim 37. Amendments have been made to claims 38 and 46 and claims 44 and 52 are restored and respectfully dependent on claims 38 and 46.

**CONCLUSION**

Having addressed all rejections and objections, Applicant respectfully submits that the subject application is in condition for allowance and a Notice to that effect is earnestly solicited. If necessary, the Commissioner for Patents is authorized to charge or credit the **Account No. 50-3102** for any deficiency or overpayment.

Respectfully submitted,

Date: November 13, 2006

By:  \_\_\_\_\_

Correspondence Address:  
Customer No. 49637

Thomas M. Isaacson  
Attorney for Applicant  
Reg. No. 44,166  
Phone: 410-286-9405  
Fax No.: 410-510-1433